Convergent evolution in silico of biochemical log-response

Mathieu Hemery, Paul François

Convergent evolution in silico of biochemical log-response

Mathieu Hemery[†,‡] and Paul François[*,,‡]

†*EPI Lifeware, INRIA Saclay, Palaiseau, France*

‡*Rutherford Physics Building, 3600 rue University, H3A2T8 Montreal, Québec, Canada*

E-mail: paulf@physics.mcgill.ca;mathieu.hemery@polytechnique.org

Abstract

Numerous biological systems are known to harbour a form of logarithmic behaviour, from Weber's law to bacterial chemotaxis. Working on a logarithmic scale allows the organism to respond appropriately to large variations in a given input at a modest cost in terms of metabolism. Here we use a genetic algorithm to evolve biochemical networks displaying a direct logarithmic response. Interestingly, a quasi-perfect log-response implemented by the same simple core network evolves in a convergent way across our different replications. The best network is able to fit a logarithm over 4 order of magnitude with an accuracy of the order of 1%. At the heart of this network, we show that a logarithmic approximation may be implemented with one single non-linear interaction, that can be interpreted either as a phosphorylation or as a ligand induced multimerization and provide an analytical explanation of the effect. Biological log-response might thus be easier to implement than usually assumed.

Introduction

Multiple cellular pathways have evolved to work over several orders of magnitude. A spectacular example can be found in immune recognition, where cells can simultaneously be insensitive to tens of thousands of non cognate ligands while being sensitive to minute concentrations of agonists with only slightly higher affinity[1] . In this case, specialized mechanisms, implicating proofreading cascades and multiple biochemical feedbacks have been described[2] , implementing a so-called adaptive proofreading strategy[3,4] . Similarly, chemotactic response in E. Coli has been shown to be logarithmic, and there too, specific biochemical adaptation pathways are responsible[5] .

Log-normal distributed concentrations are also the norm rather than the exception in cellular biology[6] and can be explained by a variety of simple mechanisms such as exponential decay or growth with a normally distributed rate or the accumulation of multiplicative noises[7] . But if cells have to deal with enormous phenotypic variability, it seems reasonable to assume that they are routinely able to take logarithms of concentrations to filter it out. This is paradoxical since most of biochemical reactions rather work on a linear scale. Classical examples are Michealis-Menten/Hill (or more generally sigmoidal) kinetics, where rates quickly saturate above a well defined threshold[8] .

This motivates a special study of general mechanisms underlying "logarithmic-response" over multiple orders of magnitude. As said above, previous works have already identified possible models and mechanisms to explain logarithmic sensing but to our knowledge few studies try to develop a direct biochemical computation of a logarithmic function. A general rule in such models is that one or several feedbacks are required to allow for logarithmic sensing. As an example, a synthetic biology approach combined positive feedback on low-copy-number-plasmid with a shunt component on high-copy-number-plasmid to provide sensitivity over 4 orders of magnitudes[9] . Another proposal relies on an allosteric system with sigmoidal responses, that modulates its saturation threshold with the help of a negative feedback to extend its range[10] . It is neither clear if such mechanisms are ubiquitous, nor if simpler

mechanisms exist.

In this work, we turn to machine-learning based approach to design biochemical networks showing a response proportional to the logarithm of a given input. Such approaches have produced very good results to recover and predict models underlying physical or biological systems[11,12] . A key advantage of those is their interpretability, circumventing the well known "black box" problem currently observed in machine learning[13] . We use φ-evo an evolutionary algorithm tailored for such biological problems[14] . We find a simple and generic mechanism generating logarithmic response, where the Input concentration catalyzes a reaction where rates are polynomial in the substrate concentration. This allows for log sensitivity over several order of magnitudes. This could be easily implemented for instance with ligand induced multimerization/oligomerization of receptors[15] , a process that does not require any feedback. Positive feedforward and negative feedback interactions can then be added to improve the logarithmic behavior over 4 orders of magnitudes. Our work reveals that there might be simple mechanism without feedback underlying log-sensing in a generic way, which therefore might be more frequent than expected.

Theoretical Methods

We use the φ-evo software to evolve biochemical networks able to implement a logarithm[14] . In short φ-evo is a Python/C-based software simulating classical Darwinian evolution on a population of gene networks. Gene networks are encoded with the help of a pre-defined biochemical grammar. Possible interactions include protein protein interactions, transcriptional regulations, catalytic interactions (possibly with strong non-linear behaviours) and both passive and enzymatic degradations. All those interactions are included in the grammar for the simulations presented here. From a given gene network, φ-evo generates a set of ordinary differential equations that are numerically integrated to determine the behavior of this individual. Those simulated gene networks then go through a cycle of selection, growth and

mutation, briefly detailed below. For an extended presentation of the biochemical grammar, modelization and algorithmic details, we refer the reader to the software description.[14]

φ-evo requires the definition of a fitness function for Darwinian selection. We use the simplest possible objective function to promote the evolution of logarithmic behaviour as depicted in figure 1. Each network contains an "Input" and an "Output" biochemical species. Then, the concentration of the Input is fixed to a random constant value I (the distribution of which is exponential between 1 and 10^4) for an interval of duration τ and 0 otherwise (the distribution of τ is the sum of a constant and an exponential term making it both broad and never too short). φ-evo then integrates the differential equations corresponding to the biochemical network, and computes as an output O of the network the maximum of the concentration of the "Output" species as a function of time [1] .

For each network, the corresponding set of ODE is then integrates $N = 100$ times with randomized choices for both I and τ and those data are collected to compute the network fitness:

$$\varphi = \frac{\sigma\left(\alpha(I,\tau)\right)}{\left\langle\alpha(I,\tau)\right\rangle^2} = \frac{\left\langle\alpha^2(I,\tau)\right\rangle}{\left\langle\alpha(I,\tau)\right\rangle^2} - 1, \text{ with: } \alpha(I,\tau) = \frac{O(I,\tau)}{\log(I)}. \tag{1}$$

Here σ indicates the variance and $\langle \dots \rangle$ the mean. During our simulation, the average is taken over the N integrations with different random inputs. Intuitively, if the network takes a perfect logarithm then $\alpha(I,\tau)$ should be a constant independent of I and τ, so that each independent input choices should give the same constant α and the standard deviation defined in equation 1 should be 0. We thus aim to minimize σ. The division in the fitness definition is such that it does not impose a predefined value for the constant in front of the logarithm definition, thus any function of the form $\alpha \log(I)$ will be selected. However, the higher the α the lower the noise and this creates an indirect selection for the highest possible constant which is the desired behaviour (see figure 1). σ can be interpreted as the

[1] those choices can be relaxed, for instance results were similar when the "output" was the integral of the output concentration or its final concentration instead of it maximum. This property is actually a direct consequence of our desired independance of the final output in the time of the output τ as discussed later.

quality of the logarithm taken as a linear function of $\log I$ (lower σ being higher quality). Note that while the value of τ appears in the computation of O, it is not present in the evaluation of the fitness φ, hence the network should be able to get rid of the time for which the input is present and respond only to the very intensity of the input signal. Figure 1 visually illustrates examples of fitness computation.

φ-evo uses an elite strategy to implement the selection. At each generation of the algorithm, it selects half of the network population based on their value of φ. Some small random component is added to φ to ensure mixing of the networks populations. Once half of the networks are selected, we duplicate them and the duplicated networks go through a round of mutations. Mutations randomly add, remove or change biochemical interactions to the networks, based on the predefined biochemical grammar. In Supplement we give initialization files, fitness definition and mutation rates allowing to reproduce results presented here. For more details on φ-evo, check Ref. 14.

Results

Convergent evolution towards a simple biochemical implementation of the logarithm

A typical evolutionary simulation including a simulated evolutionary pathway (with value of φ as a function of the number of generations) and intermediate network structures is presented in Fig. 2.

The evolution reaches a fitness lower than 10^{-2} in less than 150 generations and stabilize around 10^{-3}. This means that for such network, the average relative deviation of the Output from the logarithm of the Input (up to a multiplicative constant) is less than 5%.

The simplest network topology achieving such a performance appears at generation 134,

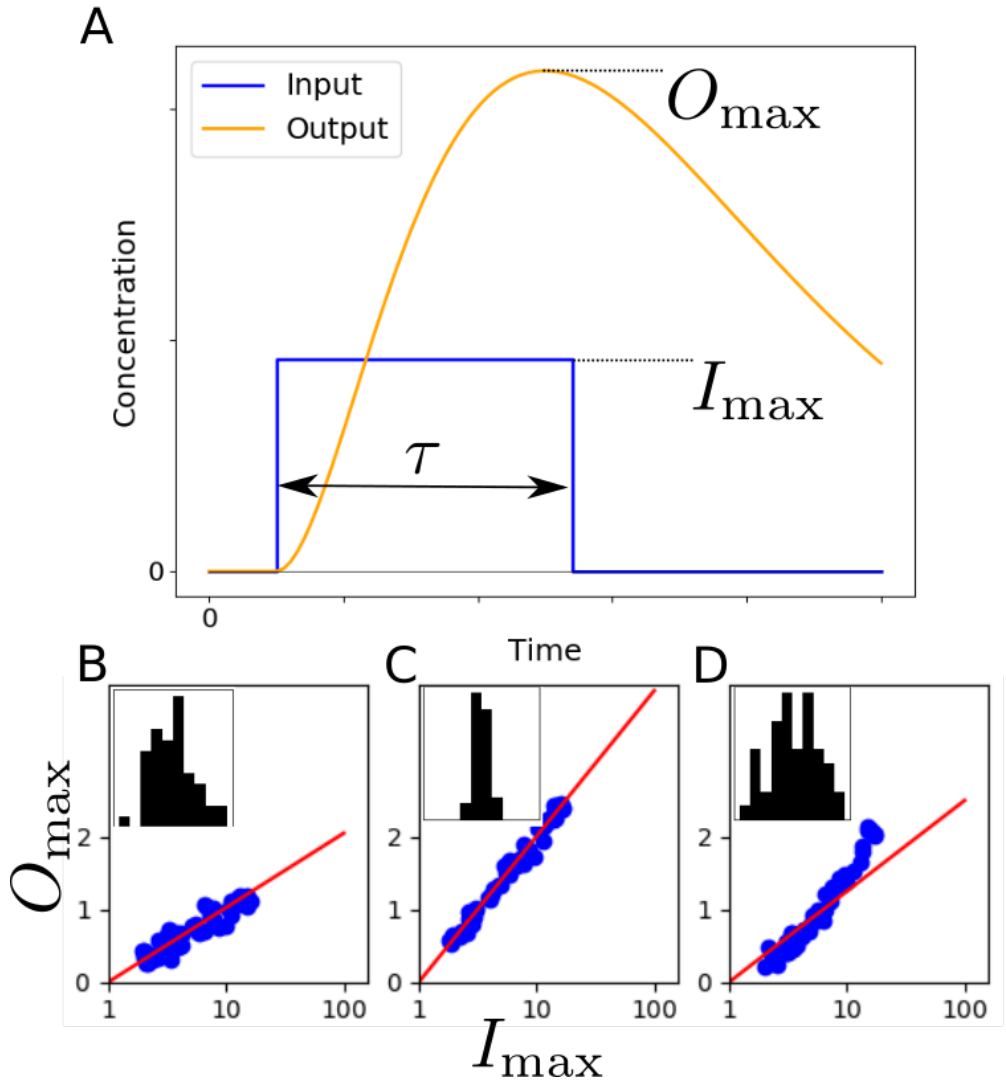


Figure 1: To compute the fitness, each network is evaluated under different randomized inputs (panel A), and the results are then gathered in a single score (panels BCD). **A**. Sketch of an input (in blue) and the response of the network in orange, the input parameters (τ and $I_{\max}$) are drawn from two distributions and the response is computed numerically through integration of the ODE corresponding to the network. We then retrieve $I_{\max}$ and $O_{\max}$ to compute the fitness score φ. **B-C-D**. Sketches of the fitness computation, blue dots represent the responses for different input on the $I_{\max}$, $O_{\max}$ plane. The red line gives the ideal linear response and the inset represent the rescaled distribution the standard deviation of which we have defined as our fitness score. (note that the scale are the same for every plot). The final score of this three cases is: $\varphi_B = 3.10^{-2}$, $\varphi_C = 6.10^{-3}$ and $\varphi_D = 7.10^{-2}$. Note that this definition also naturally favors large response range as a high mean will flatten the noise and lower the variance as seen on panel C.

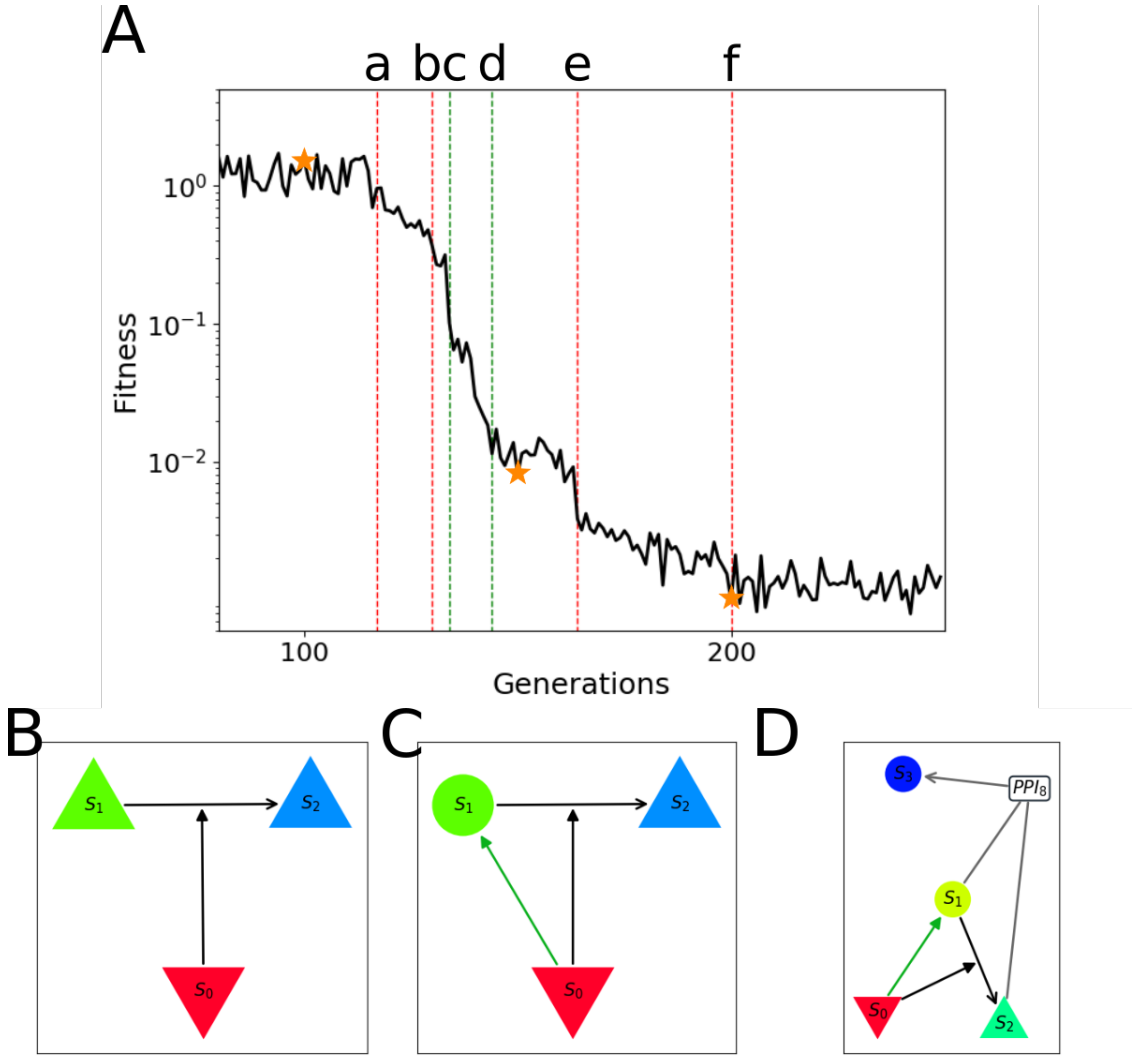


Figure 2: **A**. Excerpt of the fitness as a function of time (in generation). Vertical red lines correspond to changes of network topologies while green lines correspond to changes of major parameters. Orange stars correspond to the networks of panels B-C-D. See main text for details. **B - D**. Representation of the best network in the current population after generations 100 (B), 150 (C) and 200(D). Boxed coloured letters correspond to biochemical species, Input is the red triangle S_0, Output is the Species S_2. Arrows indicate possible biochemical interactions. In B, only a phosphorylation is present (red arrow), in C the input activates the transcription of the first species (green arrow). In D, the phosphorylated and unphosphorylated species form a complex, this interaction is an *ad hoc* form of repression that is very current in networks produced with φ-evo.

and is displayed in figure 2 C. Differential equations corresponding to this network are:

$$\frac{dS}{dt} = k_a \frac{I^{n_a}}{I^{n_a} + {t_a}^{n_a}} - k_p I \frac{S^{n_p}}{S^{n_p} + {t_p}^{n_p}} + k_d O - \delta_S S, \tag{2}$$

$$\frac{dO}{dt} = k_p I \frac{S^{n_p}}{S^{n_p} + {t_p}^{n_p}} - k_d O - \delta_O O. \tag{3}$$

This network work in the following way: the Input first activates the expression of a gene encoding a protein S (term $\frac{I^{na}}{I^{na}+{t_a}^{na}}$), then S is non-linearly catalyzed by I (term $I\frac{S^{np}}{S^{np}+{t_p}^{np}}$) to produce the output, here in the example of a phosphorylation. Note that the input acts both as a transcriptional activator and a catalyst (kinase).

Improvement of the fitness over the course of the evolution corresponds to major changes in the topology or in the parameters of the system. For the example displayed on figure 2**A**, at generation 117 (a), the network selects as a new output the catalyzed version of S. At generation 130 (b), S gets activated by the Input, so that the topology of the network of figure 2 C is now present. Generations 134 and 144 (c and d) correspond to increase in production rate of S and in catalytic rate by the Input, which allows for subsequent major improvement of the fitness, plateauing at fitness 10^{-2}. At generations 162 and 200 (e and f), changes in network topologies allow for small improvement of the fitness. For instance, at generation 200 a protein-protein interaction appears between S and O, allowing for an effective repression of S by O that slightly improves the fitness.

The motif displayed in figure 2**C**, that allow the simulation to reach low fitness region, appears repeatedly in our simulations. For instance, in a typical run of 20 consecutive simulations with exact same mutation parameters, 10 ended with a sufficient fitness to be considered as successful (of the order of 10^{-2} or lower). Surprisingly, each of these simulations display the same core topology at the heart of the network of interactions corresponding to Fig 2 C. Thus our simulations display strong convergent evolution. Variations on the numerical value of the parameters, other species and interactions around the functional core can be observed but do not impinge the main function of the core network. Interestingly,

not only the core final network is conserved, but the evolutionary path is too, with the same interactions appearing in the same order (for an example compare Fig 2 with Fig. S 1 in the Supplement). Over many simulations, the most common addition to the basic motif of Fig 2 D is a form or another of negative control of the output on its own gene (that is the species S), see e.g. Fig. 3 A.

Importantly, this particular network evolves because there is no restriction in φ-evo on the number/nature of reactions a species can be involved in. Here, the input acts both as a transcription factor and a kinase. This combination may not be possible in nature in some cases, but we checked that the logarithm function is preserved if we separate the kinase and transcriptional activities of the Input into an extra kinase activated transcriptionally by the Input (see Supplement for an explicit example).

Logarithm phenomenology and mechanism

A sketch of the best topology we obtained is displayed in Fig. 3 A, which is very similar to the network of Fig 2 C with an additional negative control of S by the Output[2].

Fig. 3 details the behaviour of the two species composing the network as a function of the Input, compared to a logarithmic behaviour. We clearly see that such a simple network can implement a precise logarithm of the Input over 4 orders of magnitude, compatible with what is typically observed in processes such as immune recognition. Furthermore, this logarithmic behavior works at steady state values of both S and O, keeping I constant. However it is far from clear how such behaviour emerges from the interactions, requiring a more detailed analysis to disentangle the influence of various biochemical interactions.

No simple analytic forms can be derived from the network equations, due to multiple non-linearities. To proceed, we successively remove parts of the networks, and further optimize parameters using our fitness function to identify which interactions are most crucial at steady

[2]The precise form of this control is suject to variability between the different simulations, it may be a transcriptional regulation, a protein protein interaction that form a dimer with low dissociation or high degradation or a direct enzymatic degradation, in all these cases however, the negative influence is manifest.

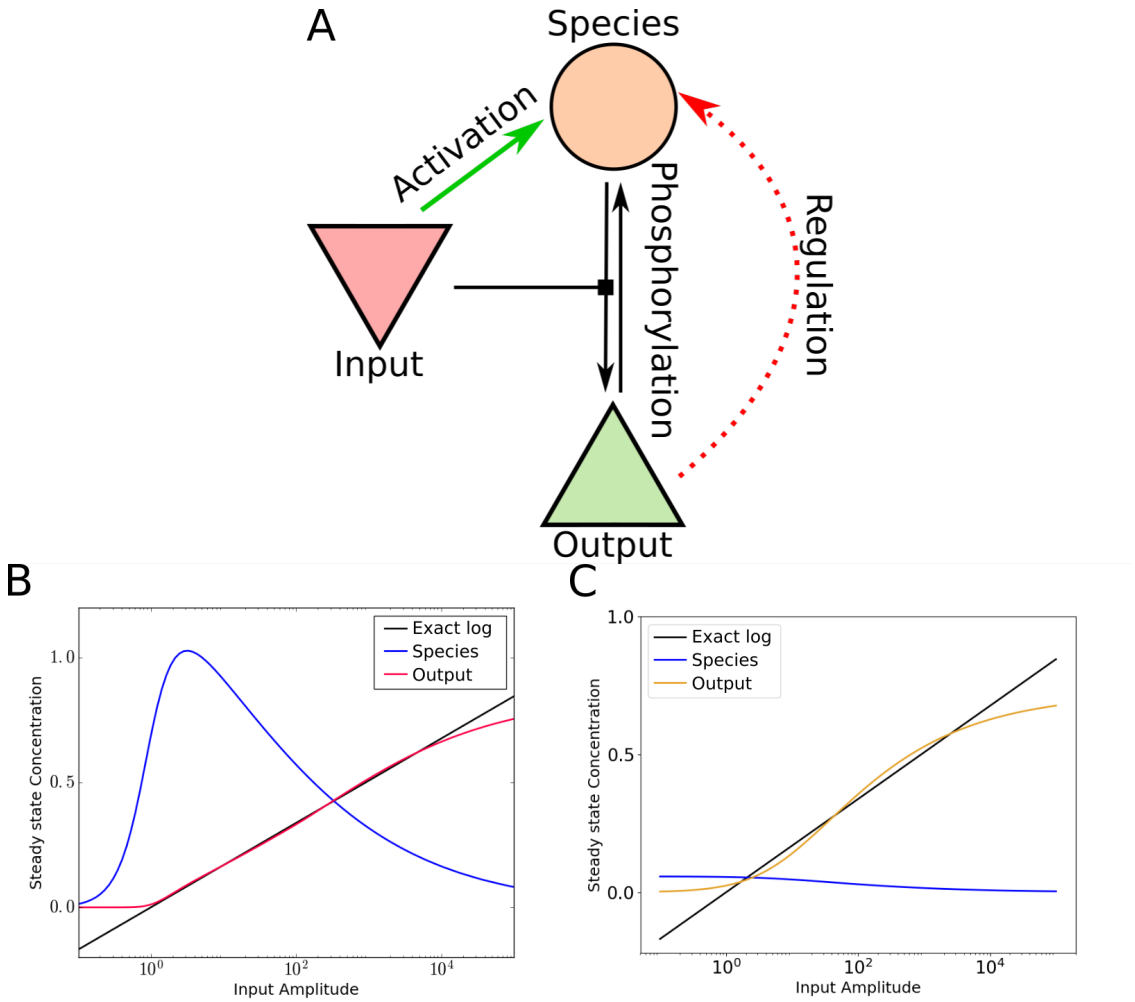


Figure 3: **A**. Convergent network structure evolved to produce a logarithmic response. Typically, the input activates or produces a species that should then be phophorylated to produce the output. To achieve an even better fitness, the evolved networks then may harbor a form or another of regulation (that is regulation, complexation or enzymatic degradation) of the output on its non-phophorylated form. **B**. Steady state concentration of the species A and B as a function of the input concentration (in log scale), the black line indicates the logarithmic function on which we fit our output concentration over the $[10^0, 10^4]$ range. Note the complex behavior of the precursor species the non-monoticity of which is crucial to perform the desired function. The mean error between the output and the desired logarithm is of the order of 1% on the range of interest. **C**. Same as **B** but for the case where both activation and regulation has been deleted. Note the weak range of variation of S and the deviation of O from the desired target. Still, the approximation of a logarithm is roughly correct, the mean error from the logarithm being of the order of 10%.

state.

As expected from the example of Figure 2, the negative regulation of species S by the Output is not crucial, and the network behaviour can be almost logarithmic without it. Further removing the activation within the networks and optimizing parameters yields a behaviour that, without being logarithmic strictly speaking, still presents strong similarity with a logarithm upon 4 orders of magnitude as depicted in Figure 3 C.

This behaviour can be analytically understood by considering the following limits of equations 2 and 3. On the one hand, at steady state for 3, assuming we are far from saturation for the catalytic interaction, we clearly have

$$O \propto IS^{n_p} \tag{4}$$

On the other hand, substituting steady state equation 3 in 2 and assuming constant production, we have

$$\delta_S S + \delta_O O = k_a \tag{5}$$

which is an effective conservation law stemming from the fact that S is transformed into O.

Now let us remark that while I varies over 4 order of magnitudes, S and O vary only on a limited range due to their logarithmic nature. From equation 4 we can write $S = \left(\frac{O}{I}\right)^{\frac{1}{n_p}}$ and thus injecting this solution in equation 5:

$$\delta_O O = k_a - \delta_S \left(\frac{O}{I}\right)^{\frac{1}{n_p}} \tag{6}$$

$$\delta_O O = k_a - \delta_S \exp\left(\frac{1}{n_p}(\log O - \log I)\right) \tag{7}$$

$$\delta_O O = k_a - \delta_S - \frac{\delta_S}{n_p}\log O + \frac{\delta_S}{n_p}\log I \tag{8}$$

$$O = \frac{k_a - \delta_S}{\delta_O} + \frac{\delta_S}{\delta_O n_p}\log I \tag{9}$$

where we Taylor expand the exponential up to first order in $\frac{1}{n_p}$ and use the fact that $\log O$ is negligible in the last expression since, as we noticed, O is small by construction while I can take large values.

Equation 9 promises us a perfect logarithm (up to first order) if $k_a = \delta_S$ so that the first term in the r.h.s. cancel. Moreover we also see that increasing n_p should be counterbalance by either lowering δ_O or increasing δ_S.

To verify this prediction, we numerically solve the equation 6 with the choice of coefficients described above and check that it effectively gives a logarithmic approximation when $n_p \to +\infty$ as shown in supplement.

Parameter dependency and range constraints

While we have identified in the previous section the core mechanism for log-sensing, it is worth examining more carefully what range of parameters give consistent logarithmic behaviour.

To further clarify parameter dependency, we turn back to the full network displayed in Fig. 3 A, and study its behaviour when parameters are randomly chosen. For 5000 parameter sets, we compute the fitness φ and the range of output defined as the difference between the maximum and the minimum value of the Output when the Input varies from 1 to 10^4 and present the results in the Fig 4. As we hope to find networks that harbour a low fitness but a high range, the better networks will be found in the bottom right corner (corresponding to maximum range, minimum fitness). The optimized network of Fig. 3 A is indicated with a red dot, and clearly displays parameter optimization with respect to networks with random parameters. Still it is interesting to see that random sets of parameters may reach interesting regions of the functional plane.

We define two subsets of interest. Blue dots identify a sub population of networks clustering on a bottom-right front. Parameter sets on this front achieve small values of the ratio between fitness and range. Small fitnesses lower than 10^{-2} with ranges of order 1 are still

reached on this front, even with parameters chosen randomly. We define a second subset of parameters with an orange window of networks with both small fitness and high range, from our 5000, 260 networks reach this threshold thus indicating that $5.2 \pm 0.6\%$ of the parameter space produced an acceptable logarithmic approximation.

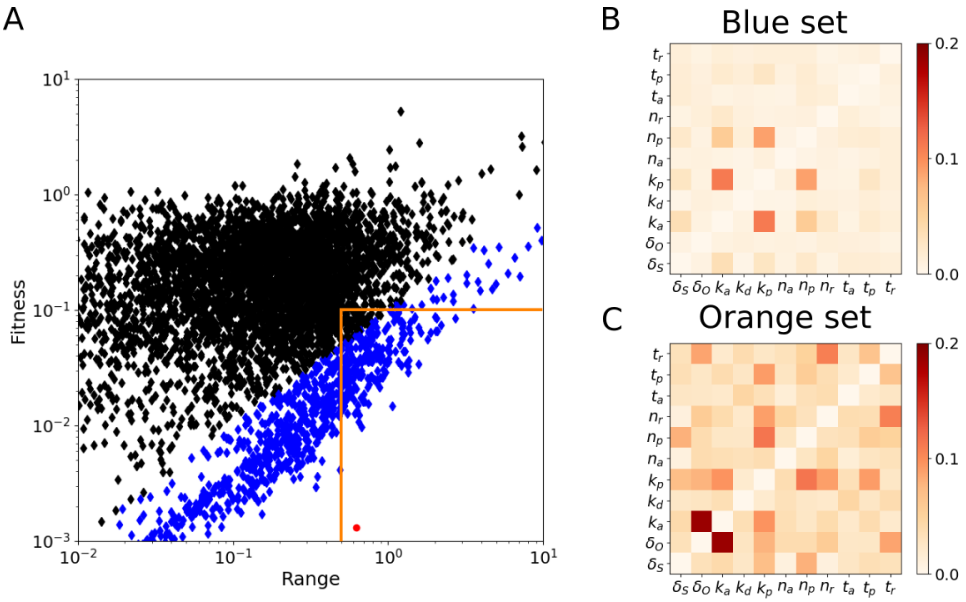


Figure 4: **A**. Fitness and range of response (defined as the difference between maximum and minimum Output values when the Input is varied) for 5000 random set of parameters (see details in Supplement for ranges of variation). Better networks are in the lower right corner. The red dot correspond to the optimized set of parameters for network of Fig. 3 A. We focus on two subsets of parameters. Blue dots are network close to minimal fitness over range ratio, corresponding to a linear front (~ 870 parameter sets), The orange window are parameter sets that have both good fitness and range (~ 260 parameter sets). **B-C**. Mutual information between parameters for the two parameter sets defined in Panel A. Warmer colours define stronger correlations, see the main text for the full detail of these correlations. The second set being smaller is also more noisy.

We now consider those two subsets of parameters to determine how parameters are constrained. Distribution of each parameter inside a set does not reveal any particular pattern (see figure 10 in supplement), which steered us to look for second order correlation.

We compute the mutual information score for all couple of parameters for those two subsets, Fig 4 B-C (see Supplement for more details). High mutual information between two parameters means that their distributions are intertwined, thus indicating correlation

between their values from set to set and revealing constraints on the biochemical interactions.

Most couples of parameters have low mutual information, indicating lack of any strong constraints between them. However, we identify three strong constraints for the network to perform well. The blue set presents two constraints. A first one is see on the activation rates: $k_p + k_a < 1$ (Fig in Supplement), that constraints the sum of the production rate of S with the maximum catalytic rate. Indeed, if those rates are too high, the system will be blocked in the saturated regime and cannot take advantage on the non-linearity of the interactions, this is thus consistent with our analytical study above.

Another less intuitive constraint is observed on the phosphorylation parameters k_p, n_p see Fig. 5 : there is basically almost no selected set in the region above the first diagonal $n_p = k_p$ (in rescaled units). To get a better understanding of the origin of this constraint, we took the network of Fig. 3 C, ran several simulations with increasing values of n_p, and k_p, and study the behaviour of S. Increases of n_p clearly moves the behaviour of S towards the right, with saturation appearing for small Input concentration. Conversely, increases of k_p moves the behaviour of S towards the left with less saturation for small Input concentration. Thus to get a linear behaviour of S over the range of concentration considered, from a given set of optimized parameters, an increase of n_p can be essentially compensated with an increase of k_p. However it is also clear that increases of n_p moves the curve more significantly than increases of k_p, so that we expect there is a limit for which increases k_p can no longer compensate n_p, explaining why the region where n_p is too big w.r. to k_p is forbidden.

The orange set allow us to identify one last strong constraint: $k_a > d_o$ asking that the degradation rate should be smaller than the production. This constraint is easy to understand as it ensures that the concentration of output may be high and thus allows for a large range in the output concentration.

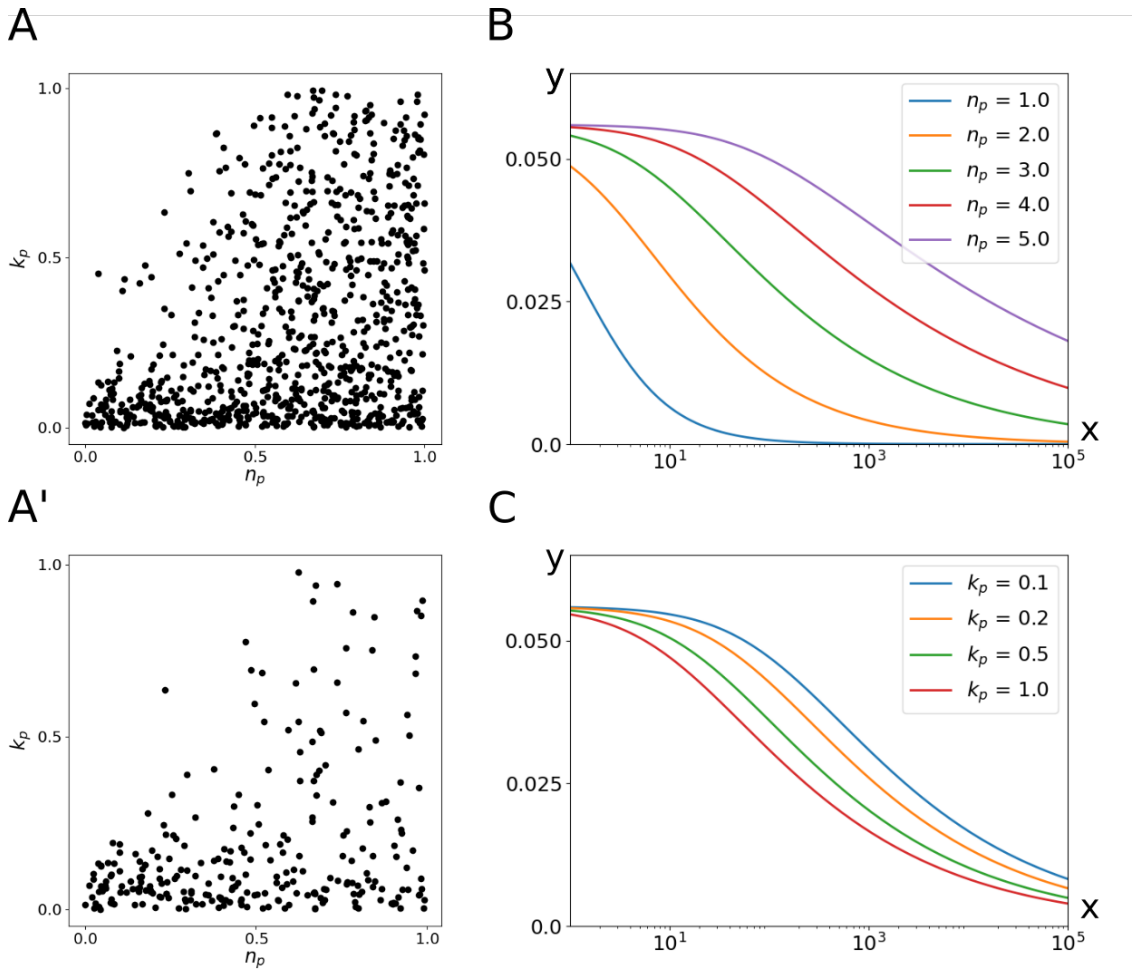


Figure 5: **A**. Relation between k_p and n_p for the network in the blue (A) and orange (A') set. The constraint is manifest although subtle. **B**. Influence of n_p on the solution of the network 3**C**, the hill coefficient of the phosphorylation thus allow to have a more and more linear response but shift the linear part on the high input regime. **C**. Influence of k_p on the solution of the network 3**C**, the rate of the phosphorylation thus correspond to a shift on the low input that allow to compensate for the shift due to the hill coefficient. Note however that this shift seems logarithmic in k_p (similar shift for our 4 curves that have a log spacing).

Discussion

We have used in silico evolution (the φ-evo algorithm) to design a biochemical log-sensor. Results of our simulations show that an almost perfect logarithmic sensor over 4 orders of magnitude can be obtained by combining feedback and feedforward interactions with a non linear interaction catalyzed by the Input.

The key log-sensitivity comes from Eq. 6 relating I and O. It is clear here that such non-linear catalysis could arguably constitute one of the simplest (if not the simplest) mechanism implementing a logarithmic response. Such process is not unrealistic. A first possibility is to have multiple phosphorylations as suggested in REF. Another easy way to get high powers in Eq. 4 is to assume that biochemical species S have to form multimers before being catalyzed in the active form O by I. Some Input can then trigger internalization of receptors and subsequent signalling, effectively implementing log-sensing. Such processes are common in biology. For instance in G protein-coupled receptors signalling, agonist triggers oligomerizations of receptors[15] . Another context is immunology where cellular receptors are well-known to form multimers at the surface of the cell[16] and are phosphorylated to trigger response[17] . Such networks (with internal feedback) could thus work directly in the log range for Input concentration, ensuring broader response or non-trivial scaling laws[18] .

More generally, assuming a relation such as Eq. 4 holds irrespective of specific biochemistry, we can expect O and S to vary logarithmically with I as long as both O and S are constrained to vary on a predefined range. Here, the latter constraint stems from the effective conservation law in equation 5, which imposes limits on both concentrations at the same time. An alternative way to perform this would simply be that neither S and O are degraded, nor S is produced, so that Eq. 4 would directly be mass conservation. We have also identified parameter dependencies for this system to work correctly, key of which is the necessity to play on the non linearity of the hill functions, that is avoid both high and low saturation regimes. The constraint that n_p should be small enough with respect to k_p also suggests that the non-linearity should not be too high. This is of biological relevance

in a case where n_p is the number of receptors forming multimers, which thus should stay relatively low.

One can also refine the system to get a "perfect" log, by addition of other feedbacks that are found by φ-evo. Of course, it is not entirely clear why a cell would need to take an exact log, versus a response that would only be log-like. In our case, extra feedbacks can rather be thought of as post-processing of a core signal sensitive over orders of magnitudes, and as such can be likely performed otherwise downstream of the Output. This vision is very different from previous proposals, where feedback need to be carefully engineered to ensure log sensitivity[9,10] . While it is not clear which solution is more amenable to be found by biological evolutionary tinkering, solutions presented here are *de facto* evolvable, within the range of allowed biochemical interactions and mutations in our simulations. The dual fact that we observe very strong convergent evolution towards this solution and that receptor multimerization is a possible common mechanism to implement it suggests that the solution we propose might be easier to find by evolution, and therefore potentially ubiquitous.

Supporting Information Description

- A short document gathering figures, details and explanations not important enough to be embedded in the main text and refer in it as supplement.
- Initialization files of the main run described in the text (to be used with φ-evo software).

Note also that all our source code files for the optimization or simulation are written in Python and may be obtained through mailing the authors.

Acknowledgement

References

(1) Feinerman, O.; Germain, R. N.; Altan-Bonnet, G. *Molecular Immunology* **2008**, *45*, 619–631.

(2) Altan-Bonnet, G.; Germain, R. N. *PLoS Biology* **2005**, *3*, e356.

(3) François, P.; Voisinne, G.; Siggia, E. D.; Altan-Bonnet, G.; Vergassola, M. *Proc Natl Acad Sci U S A* **2013**, *110*, E888–97.

(4) Lalanne, J.-B.; François, P. *Physical Review Letters* **2013**, *110*, 218102.

(5) Kalinin, Y. V.; Jiang, L.; Tu, Y.; Wu, M. *Biophysj* **2009**, *96*, 2439–2448.

(6) Milo, R.; Phillips, R. *Cell Biology by the Numbers*; Garland Science, 2015.

(7) Koch, A. L. *Journal of Theoretical Biology* **1966**, *12*, 276–290.

(8) Frank, S. A. *Biology Direct* **2013**, *8*, 221.

(9) Daniel, R.; Rubens, J. R.; Sarpeshkar, R.; Lu, T. K. *Nature* **2013**, *497*, 619–623.

(10) Olsman, N.; Goentoro, L. *Proc Natl Acad Sci U S A* **2016**, *113*, E4423–30.

(11) Daniels, B. C.; Nemenman, I. *PLoS ONE* **2015**, *10*, e0119821.

(12) François, P. *Seminars in cell & developmental biology* **2014**, *35*, 90–97.

(13) Knight, W. There's a big problem with AI: even its creators can't explain how it works. `https://www.technologyreview.com/s/604087/the-dark-secret-at-the-heart-of-ai/`.

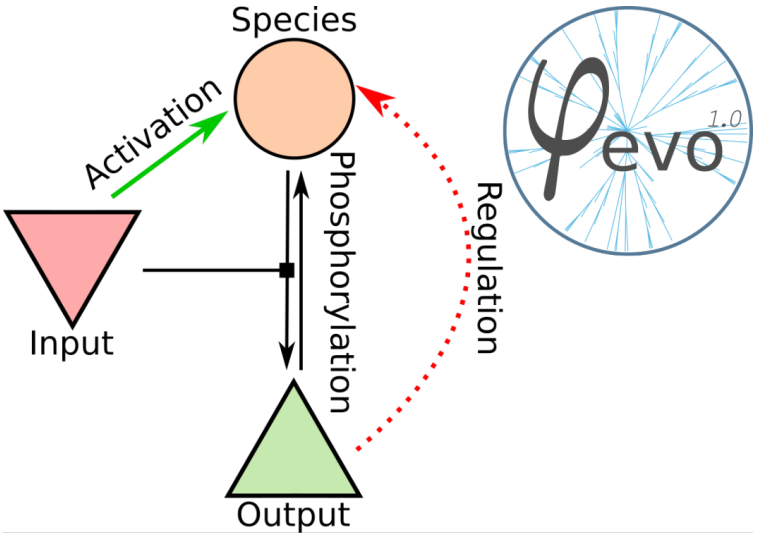
(14) Henry, A.; Hemery, M.; François, P. *PLoS computational biology* **2018**, *14*, e1006244.

(15) Li, Y.; Shivnaraine, R. V.; Huang, F.; Wells, J. W.; Gradinaru, C. C. *Biophysical Journal* **2018**, *115*, 881–895.

(16) Taylor, M. J.; Husain, K.; Gartner, Z. J.; Mayor, S.; Vale, R. D. *Cell* **2017**,

(17) Kersh, E. N.; Shaw, A. S.; Allen, P. M. *Science* **1998**, *281*, 572–575.

(18) Tkach, K. E.; Barik, D.; Voisinne, G.; Malandro, N.; Hathorn, M. M.; Cotari, J. W.; Vogel, R.; Merghoub, T.; Wolchok, J.; Krichevsky, O.; Altan-Bonnet, G. *eLife* **2014**, *3*, e01944.



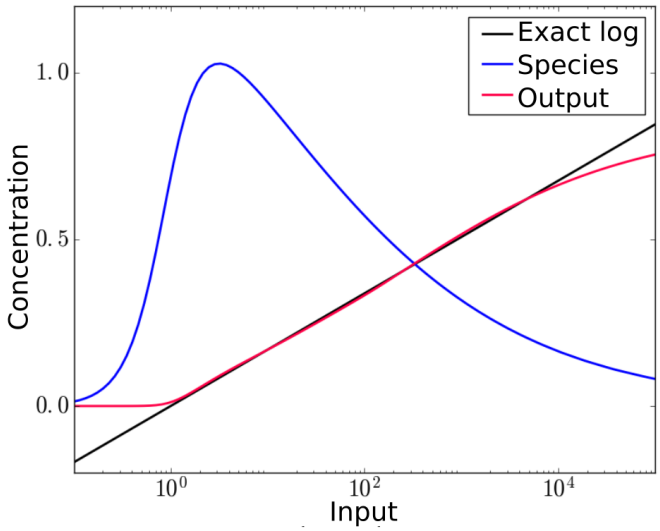


Figure 6: TOC figure